Multi-class Semi-supervised SVMs with Positiveness Exclusive Regularization

Xiaobai Liu[†,‡], Xiaotong Yuan[‡], Shuicheng Yan[‡], Hai Jin[†]
† Huazhong University of Science and Technology, Wuhan, China
‡ National University of Singapore, Singapore

Abstract

In this work, we address the problem of multi-class classification problem in semi-supervised setting. A regularized multi-task learning approach is presented to train multiple binary-class Semi-Supervised Support Vector Machines (S3VMs) using the one-vs-rest strategy within a joint framework. A novel type of regularization, namely Positiveness Exclusive Regularization *(PER), is introduced to induce the following prior: if an unlabeled sample receives significant positive response from one of the classifiers, it is less likely for this sample to receive positive responses from the other classifiers. That is, we expect an exclusive relationship among different S3VMs for evaluating the same unlabeled sample. We propose to use an $\ell_{1,2}$-norm regularizer as an implementation of PER. The objective of our approach is to minimize an empirical risk regularized by a PER term and a manifold regularization term. An efficient Nesterov-type smoothing approximation based method is developed for optimization. Evaluations with comparisons are conducted on several benchmarks for visual classification to demonstrate the advantages of the proposed method.*

1. Introduction

Multi-class classification can be considered the simplest and most natural learning problem going beyond the binary setting. It is possible, and still common practice, to reduce multi-class learning to a set of binary classification problems. Crammer and Singer [7] provide evidence that, in general, genuine multi-class approaches can be superior, but other studies suggest that it is hard to beat the one-vs-rest heuristic, both in accuracy and computational complexity. Particularly, multi-class Support Vector Machines (m-SVMs) have been investigated in many practical situations for several years, both in dual form [10] and in primal form [6]. Although widely applied in practice, one limitation for m-SVMs is that the individual classifiers are often trained independently, ignoring potential correlation among the classifiers.

Multi-task learning (MTL) is a statistical learning framework which targets at learning different models such as SVMs in a joint manner. When there are relations between the tasks to learn, it can be advantageous to learn all tasks simultaneously instead of following the more traditional approach of learning each task independently. There has been a lot of experimental work showing the benefits of such multi-task learning relative to individual task learning when tasks are related, see [1, 5, 8, 19].

In this paper, we develop a novel approach for learning multi-class semi-supervised SVMs (m-S3VMs) [1] within a framework of regularized MTL. The motivation is to jointly learn one-vs-rest S3VMs for individual classes by imposing regularization terms which enforce certain desired correlation across different classifiers. To the best of our knowledge, this is the first attempt to formulate m-S3VMs as a regularized MTL problem. The points in the next subsection highlight two key contributions of our method.

1.1. Our Contributions

The first contribution is a novel type of discriminative regularization, the PER for *positiveness exclusive regularization*, which is defined on the outputs of individual classifiers on unlabeled samples. The principle of PER is that if a sample receives significant positive score from one of the classifiers, we expect that it is less likely for this sample to receive positive scores from the other classifiers. Taking multi-class object recognition as an example, by assuming that the object contained in an image only belongs to one category, it is reasonable to expect that only one of the classifiers will output positive score when evaluating this image. Motivated by exclusive Lasso model in MTL [25], we propose to use an $\ell_{1,2}$-norm type regularizer as an implementation of PER.

As a second contribution, in order to incorporating PER with m-S3VMs, we propose to training m-S3VMs in primal form and formulate the problem as a regularized MTL framework. As pointed out by Chapelle [6] that training SVMs in primal, both for linear and non-linear cases, is

[1] We hope that the term S3VMs used here will not be confused with one special semi-supvervised SVMs method also called as S3VMs [4].

as efficient and accurate as in dual form. Moreover, the primal form reaps the advantage of directly incorporating additional penalties like PER with objective. These merits motivate us to adopt the primal form for m-S3VMs training. The objective is to minimize an empirical risk regularized by a PER term to impose negative correlation among tasks and a manifold regularization term to enforce geometric smoothness. The problem is convex but highly non-smooth due to the max-structure of PER. We develop an efficient Nesterov-type smoothing approximation method [16] for optimization. A number of experiments on several vision benchmarks support our observation, showing that performances are boosted by the proposed regularizer. Last but not least, the proposed PER regularizer and related optimization techniques are general and easily extendable for multi-class algorithms spanning the range from totally unsupervised to supervised learning with noisy supervision.

1.2. Related Work

Recently, there have been a lot of interests around MTL, both in theory and practice. The idea behind this paradigm is that, when the tasks to be learned are similar enough or are related in some sense, it may be advantageous to take into account these relations between tasks. Several works have experimentally highlighted the benefit of such a framework [5]. In general, MTL can be addressed through a regularization framework [8]. For example, the joint sparsity regularization favors to learn a common subset of features for all tasks [1, 19], while the exclusive sparsity regularization is used in [25] for exclusive feature selection across tasks. Our method follows the regularized MTL framework. In contrast to the existing regularization that is only model parameters dependent, our proposed regularization is characterized by data as well as model parameters, and thus is much more informative.

During the recent years, semi-supervised learning (SSL) has attracted considerable attention since it can make use of unlabeled data as well as labeled examples. A large body of SSL methods have been proposed in the past literatures, among which some popular ones include generative models like GMMs [17], transductive SVMs [12] and S3VMs [4], and a variety of graph based methods [3]. For a complete survey of SSL methods, we refer the readers to [26] and the references therein. Among others, the manifold regularized method [3] (with SVMs being the particular model taken for implementation) is the most successful SSL method which utilizes the manifold geometric regularization to improve the performance of learning on labeled data. In this work, we are interested in the training of multi-class SVMs in the setting of SSL.

The rest of this paper is organized as follows. Section 2 describes the problem formulation whereas Section 3 discusses related optimization techniques. Extensive experimental results and analysis are presented in Section 4, and finally, Section 5 concludes the paper and remarks the future work.

2. Problem Formulation

In this section, we first introduce the setup and overview of our method to m-S3VM learning, and then describe in detail the proposed PER as well as other components in a unified formulation.

2.1. Setup and Overview of Framework

We consider the following setup. Assume there is a total number of of M classes, each class is provided with a set of positive samples. We take the one-vs-rest strategy to build the training set for each class, that is, for class m we collect all the remaining samples from the rest classes as negative samples. Denote the labeled samples as $\{(x_i, y_i^m), i = 1, \ldots, l, m = 1, \ldots, M\}$ where $y_i^m \in \{+1, -1\}$ indicates whether $x_i \in \mathbb{R}^d$ belongs to the m-th class. Also we are given u unlabeled samples $\{x_{l+1}, ..., x_{l+u}\}$ and let $n = l + u$. We put both the labeled and unlabeled samples within S and arrange all the samples such that the top l samples are labeled ones.

With the labeled and unlabeled data, our goal is to train M binary-class classifiers: $f^m(x|w^m, b^m) = \langle w^m, x\rangle + b^m$ where $w^m \in R^d$ is the desired hyperplane parameter vector for class m and b^m is the bias term. Herein, we consider the nonlinear SVMs with a kernel function $k(\cdot, \cdot)$ and an associated Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$. The well known Representer Theorem [13] states that the optimal f^m exists in $\mathcal{H}$ and can be written as a linear combination of kernel functions evaluated at the training samples. Thus, for class m, we seek for a classification function with parameter vector $\alpha^m \in \mathbb{R}^n$,

$$f^m(x|\alpha^m) = \sum_{i=1}^{n} \alpha_i^m k(x_i, x). \tag{1}$$

The key idea of our method is to formulate the m-S3VM problem within the MTL framework. This is achieved by minimizing an empirical risk regularized by a manifold regularization term and a cross-task regularization term. The formulation includes three component objectives. The *first* one is the function norm regularized empirical risk suffered from the individual classifiers on labeled data. The *second* objective is a manifold regularization term [2, 3] which exploits the unlabeled samples to adjust the target functions of individual classifiers learned from the labeled data. The *third* objective is a cross-task regularization term which imposes the positiveness exclusive prior on classifiers when evaluating the same unlabeled sample. While the former two objectives are used for individual tasks learning, the last one models the correlation among tasks. Finally, by combining the above three antagonist objectives within one single objective function, we obtain a regularized MTL framework within which m-S3VMs is trained in a joint manner.

2.2. Obj-I: Empirical Risk

Denote K the kernel matrix with $K_{ij} = k(x_i, x_j)$ and K_i the i-th column of K. We consider the following function norm regularized primal empirical risk for f^m:

$$E(\alpha^m) := \sum_{i=1}^{l} V(y_i^m, K_i^T \alpha^m) + \frac{\gamma^a}{2} \alpha^{mT} K \alpha^m, \qquad (2)$$

where $V(u, v) := \frac{1}{2}[1 - uv]_+^2$ is the square of hinge loss with $[\cdot]_+$ denoting the operation of $\max\{0, \cdot\}$, γ^a is the weight of the function norm, $\|f^m\|_{\mathcal{H}} = \alpha^{mT} K \alpha^m$, in the RKHS, that enforces smoothness on the possible solutions.

2.3. Obj-II: Manifold Regularization

As is well known that the manifold regularization approach [2, 3, 11] exploits the geometry of the marginal distribution underlying the training data. The scores of two points that are close in the intrinsic geometry should be similar and vice versa. This assumption is enforced in the learning process by an intrinsic regularizer that is empirically estimated from the points cloud of labeled and unlabeled samples using the graph Laplacian associated to them. Formally, denote L the graph Laplacian associated to $\mathcal{S}$, given by $L = D - W$, where W is the affinity matrix of the data graph and D is the diagonal matrix with the degree of each node, i.e., $D_{ii} = \sum_{j=1}^{n} W_{ij}$. The following manifold regularization term measures the smoothness of f^m on graph:

$$\Psi(\alpha^m) := \frac{1}{2} \alpha^{mT} K L K \alpha^m. \qquad (3)$$

The manifold regularized SVMs is widely known as LapSVMs [3].

2.4. Obj-III: Positiveness Exclusive Regularization

We propose to use a novel type of regularization term, positiveness exclusive regularization (PER), to enforce discriminative nature of classifiers on unlabeled data. The principle of PER is: if a sample x_i receives significantly large positive score from one of the classifiers, it is less likely for x_i to receive significant positive scores from the other classifiers. Such type of regularization is natural since x_i only belongs to one of the M classes as assumed in this work, and thus it is reasonable to expect that the elements in the positive score vector $\left[[K_i^T \alpha^1]_+, ..., [K_i^T \alpha^M]_+\right]$ are exclusively to be positive. Let $\tilde{\alpha} = [\alpha^1; ...; \alpha^M] \in \mathbb{R}^{nM}$. As an implementation of PER, we introduce the following regularization function on $\tilde{\alpha}$:

$$\Omega(\tilde{\alpha}) := \frac{1}{2} \sum_{i=l+1}^{n} \|[\tilde{K}_i \tilde{\alpha}]_+\|_1^2, \qquad (4)$$

where

$$\tilde{K}_i = \begin{bmatrix} K_i^T & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & K_i^T & \cdots & \mathbf{0} \\ \mathbf{0} & \cdots & K_i^T & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & K_i^T \end{bmatrix} \in \mathbb{R}^{M \times nM}. \qquad (5)$$

The $\Omega(\tilde{\alpha})$ is essentially a kind of exclusive sparse inducing regularization which has been recently studied in both machine learning [25] and signal processing [14]. Theoretical analysis [25] shows that the $\ell_{1,2}$-norm penalty $\|\cdot\|_1^2$ encourages the variables to be exclusively selected in the output. Therefore, by using $\Omega(\tilde{\alpha})$, we expect that the scores produced by different classifiers on the same sample will be exclusively positive.

2.5. A Regularized MTL Formulation

We are now in the position to formulate m-S3VMs as a regularized MTL framework by composing (2), (3) and (4)

$$\boxed{\text{-2} \qquad F(\tilde{\alpha}) := J(\tilde{\alpha}) + \gamma^b \Lambda(\tilde{\alpha}) + \gamma^c \Omega(\tilde{\alpha}). \qquad (6)}$$

where

$$J(\tilde{\alpha}) := \sum_{m=1}^{M} E(\alpha^m), \quad \Lambda(\tilde{\alpha}) := \sum_{m=1}^{M} \Psi(\alpha^m).$$

As aforementioned, Equation (6) formulates a MTL framework with M tasks, each of which learns a binary S3VMs. The PER regularizer $\Omega(\tilde{\alpha})$ models the exclusive relationship across tasks. Through this formulation, the parameters $\tilde{\alpha}$ can be learned in a joint way. It is straightforward to verify that the objective $F(\tilde{\alpha})$ in (6) is convex but non-smooth since all the three components are convex whereas $\Omega(\tilde{\alpha})$ is non-smooth. We will develop in the next section an efficient method to optimize problem (6). Once the optimal parameter $\tilde{\alpha}^* = [\alpha^{1*}; ...; \alpha^{M*}]$ is obtained, the classification decision is given by

$$\hat{m} = \arg\max_{m} f^m(x|\alpha^{m*}).$$

2.6. Discussions

The MTL framework stated in (6) belongs to the wide class of regularized MTL (R-MTL) methods [8]. Some recent advances in R-MKL include the joint sparse MKL [1, 19] which selects a common set of features shared across the tasks and the exclusive Lasso [25] which performs exclusive selection of features across tasks. Most existing R-MTL methods utilize certain norm of classifier parameters to enforce desired properties, without explicitly involving data information in the regularization term. Differently, our PER is defined on the output scores of classifiers on unlabeled data, and thus the data information is utilized.

It is worthy to note that the PER regularization term $\Omega(\tilde{\alpha})$ and the manifold regularization term $\Lambda(\tilde{\alpha})$ are both data dependent, but favor different priors. For each m, the manifold regularization $\Psi(\alpha^m)$ encourages output clouds $\{f^m(x_i|\alpha^m)\}_{i=1}^n$ on the labeled and unlabeled data to be smooth on the graph. For each sample x_i, the PER $\Omega(\tilde{\alpha})$ encourages the output scores from the individual classifiers $\{f^m(x_i|\alpha^m)\}_{m=1}^M$ to be exclusively positive. These two regularization terms are complementary to each other. Specially, when $\gamma^c = 0$, problem (6) reduces to the a multi-task LapSVMs [3] problem where the tasks are independent with each other. Although taking LapSVMs as a special implementation of our method, we emphasize that PER can be easily combined with other S3VMs such as transductive SVMs [12] for multi-class multi-task learning.

3. Optimization

The highly non-smooth structure of $\Omega(\tilde{\alpha})$ makes the optimization of problem (6) a non-trivial task. The general purpose subgradient method as used in [25] is applicable but it typically ignores the structure of problem and suffers from slow rate of convergence. Our idea for optimization is to approximate the original non-smooth objective by a smooth function and then solve the latter by utilizing some off-the-shelf fast algorithms. In this section, we derive a Nesterov's smoothing optimization method [16] to achieve this purpose.

3.1. Smoothing Approximation

The following result founds the base of our smoothing approximation method.

Proposition 1. *For any vector $p \in \mathbb{R}^n$, there is a constant vector $v \in \mathbb{R}^n$ such that $\|[p]_+\|_1$ has a max-structure representation in the following form,*

$$\|[p]_+\|_1 = \max_{0 \leq v \leq 1} \langle p, v \rangle,$$

where $0 \leq v \leq 1$ is the element-wise constraints.

The proof is given in Appendix 1. Based on the proceeding proposition and the smoothing approximation techniques originally from [16], function $\Omega(\tilde{\alpha})$ can be approximated by the following smooth function

$$\Omega_\mu(\tilde{\alpha}) = \frac{1}{2} \sum_{i=1}^{l} q_{i,\mu}^2(\tilde{\alpha}),$$

where

$$q_{i,\mu}(\tilde{\alpha}) := \max_{0 \leq v_i \leq 1} < \tilde{K}_i \tilde{\alpha}, v_i > -\frac{\mu}{2}\|v_i\|_2^2. \tag{7}$$

Herein, μ is a parameter to control the approximation accuracy. For a fixed $\tilde{\alpha}$, denote $v_i(\tilde{\alpha})$ the unique minimizer of (7). It is standard that $v_i(\tilde{\alpha}) = \min\left\{1, \max\left\{0, \frac{\tilde{K}_i\tilde{\alpha}}{\mu}\right\}\right\}$ where operators $\max\{\cdot,\cdot\}$ and $\min\{\cdot,\cdot\}$ are taken in element-wise for the involved vectors.

3.2. On Approximation Accuracy

Denote $\|A\|_p := \max_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p}$ the induced p-norm of a matrix A. Assume that there exists a bounded feasible set of interest Q such that $\alpha^m \in Q$ for all m. Then we have the following result on approximation accuracy of Ω_μ to Ω:

Proposition 2. *$\Omega_\mu(\tilde{\alpha})$ is a μ-accurate approximation to $\Omega(\tilde{\alpha})$, that is*

$$\Omega_\mu(\tilde{\alpha}) \leq \Omega(\tilde{\alpha}) \leq \Omega_\mu(\tilde{\alpha}) + \mu M^2 R \sum_{i=1}^{n} \|\tilde{K}_i\|_1, \tag{8}$$

where $R := \max_{\alpha^m \in Q} \|\alpha^m\|_1$.

The proof is given in Appendix 2. Proposition 2 shows that for $\mu > 0$ the function Ω_μ can be seen as a uniform smooth approximation of the regularization function Ω.

To apply *optimal* first-order methods for optimization, we need to further establish the differentiability and Lipschitz continuity of $\Omega_\mu(\tilde{\alpha})$, as shown in the following theorem:

Theorem 1. *Function $\Omega_\mu(\tilde{\alpha})$ is well defined, convex and continuously differentiable. Moreover, its gradient*

$$\nabla\Omega_\mu(\tilde{\alpha}) = \sum_{i=1}^{n} q_{i,\mu}(\tilde{\alpha})(\tilde{K}_i^T v_i(\tilde{\alpha}))$$

is Lipschitz continuous with the constant

$$L_{\Omega_\mu} = M \sum_{i=1}^{n} \|\tilde{K}_i\|_2^2 \left(\frac{\|\tilde{K}_i\|_1 R}{\mu} + 1\right). \tag{9}$$

The proof is given in Appendix 3.

3.3. Smooth Minimization

Fix a small $\mu > 0$, we are going to minimize the following smooth objective:

-2 $$F_\mu(\tilde{\alpha}) := J(\tilde{\alpha}) + \gamma^b \Lambda(\tilde{\alpha}) + \gamma^c \Omega_\mu(\tilde{\alpha}). \tag{10}$$

By Proposition 2 and Theorem 1 we have that the objective function $F_\mu(\tilde{\alpha})$ is a μ-accurate approximation to $F(\tilde{\alpha})$ and it is differentiable with gradient

$$\nabla F_\mu(\tilde{\alpha}) = \nabla J(\tilde{\alpha}) + \gamma^b \nabla\Lambda(\tilde{\alpha}) + \gamma^c \nabla\Omega_\mu(\tilde{\alpha}), \tag{11}$$

where

$$\nabla J(\tilde{\alpha}) = \left[\nabla E(\alpha^1); ...; \nabla E(\alpha^M)\right],$$

$$\forall m, \nabla E(\alpha^m) = \sum_{i=1}^{n} -y_i^m K_i \left[1 - y_i^m K_i^T \alpha^m\right]_+ + \gamma^a K \alpha^m,$$

and

$$\nabla \Lambda(\tilde{\alpha}) = \left[\nabla \Psi(\alpha^1); ...; \nabla \Psi(\alpha^M)\right],$$

$$\forall m, \nabla \Psi(\alpha^m) = KLK\alpha^m.$$

It is straightforward to verify that $\nabla J(\tilde{\alpha})$ and $\nabla \Lambda(\tilde{\alpha})$ are Lipschitz continuous with constants $L_J = \|KK^T\|_2 + \gamma^a \|K\|_2$ and $L_\Lambda = \|KLK\|_2$, respectively. Therefore we get that ∇F_μ is Lipschitz continuous with constant

$$L_{F_\mu} = L_J + \gamma^b L_\Lambda + \gamma^c L_{\Omega_\mu}. \tag{12}$$

In light of the above discussion, we can employ the Accelerated Proximal Gradient (APG) method [20] to optimize $F_\mu(\tilde{\alpha})$. The optimization procedure is formally described in Algorithm 1. For a fixed μ, it is well known that APG has the optimal rate of convergence $O(1/t^2)$ where t is the iteration number. In terms of the desired residues, i.e., $|F_\mu - \min F_\mu| \leq \epsilon$, by choosing $\mu \approx \epsilon$ we have that the rate of convergence is $O(1/\epsilon)$.

```
Inputs : $K \in \mathbb{R}^{n \times n}$,$\gamma^a$, $\gamma^b$, $\gamma^c$, $\mu$, $L \in \mathbb{R}^{n \times n}$,
         $\{y_i^m, i = 1, ..., l, m = 1, ..., M\}$.
Output: $\tilde{\alpha} = [\alpha^1; ...; \alpha^M]$.
Initialization: Calculate $L_{F_\mu}$ by (12). Initialize
$\tilde{\alpha}_0, \tilde{\beta}_0 \in \mathbb{R}^{nM}$, and let $\lambda_0 \leftarrow 0$, $t \leftarrow 0$.
repeat
    $\tilde{u}_t = (1 - \lambda_t)\tilde{\alpha}_t + \lambda_t \tilde{\beta}_t$,
    Calculate $\nabla F_\mu(\tilde{u}_t)$ according to (11),
    $\tilde{\beta}_{t+1} = \tilde{\beta}_t - \frac{1}{\alpha_t L_{F_\mu}} \nabla F_\mu(\tilde{u}_t)$,
    $\tilde{\alpha}_{t+1} = (1 - \lambda_t)\tilde{\alpha}_t + \lambda_t \tilde{\beta}_{t+1}$,
    $\lambda_{t+1} = \frac{2}{t+1}$, $t \leftarrow t + 1$.
until Converges;
```

Algorithm 1: Smooth minimization for m-S3VMs with objective (10)

4. Experiments

We evaluate the effectiveness of our proposed method over two image classification problems: face recognition and multi-class object categorization.

4.1. Baselines and Parameters

As a baseline reference for the performances on multi-class categorization tasks, we use two popular regularized classifiers, SVM in the supervised setting [2] and LapSVM in semi-supervised setting [3]. In implementation of Algorithm 1, we set both γ^b and γ^c to be zero to obtain the totally supervised multi-class SVMs (**m-SVMs**) method, or set γ^c to be zero to train the multi-class LapSVMs (**m-LapSVMs**) classifiers. For each specific evaluation database, we also compare with the results reported in past literature. We implemented the optimization procedure using Matlab 2010 on a 2.63 Ghz machine with 8GB of memory. Figure 1 shows one typical convergence curve of Algorithm 1 where the X-axis indicates the increased iteration steps and the Y-axis indicates the objective function values. The data are from the YALEB dataset [9]. We fix the maximum iteration steps to be $10,000$ in following experimetns.

In evaluation, we divide the dataset used into labeled, validation and unlabeled sets. m-SVMs method uses the first two subsets for training and the last one for testing. m-LapSVMs and PER regularized m-LapSVMs shall use both the labeled and unlabeled for training. We evaluate the results using *classification accuracy* and all presented results have been obtained by averaging them on multiple randomly generated splits of the available data. A 4-fold cross-validation has been performed on the validation subset to determine the free parameters for each dataset. Particularly, the optimal weights parameters γ^a,γ^b and γ^c are determined by varying the values on the grid $\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-1}, 1, 2, 4, 6, 8, 10\}$ and chosen with respect to validation errors. Note that the comparison results of various algorithms in the rest of this section are usually consistent while using different parameter configurations and thus do not produce any effect on our claims or observations.

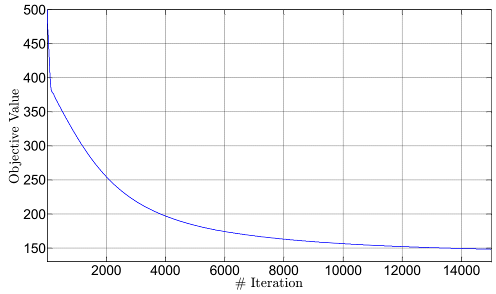

Figure 1. Convergence curve. The data used here is from the YALEB dataset [9].

4.2. Exp-I: Results on YaleB Dataset

We conduct the face recognition experiment on the Extended Yale Face (YaleB) database [9] to illustrate the advantages of our proposed formula and regularizer. It contains 16128 images of 38 human subjects under 9 poses and 64 illumination conditions. The size of each cropped gray-level image is 32×32 pixels. We use 64 near frontal face images for each individual, and simply concatenate all the gray-level pixel values to form image features, as in previous works [9, 23]. For each individual class, we randomly select $\{5, 10, 20, 30, 40, 50\}$ images for training, and the rest images for test. The reported mean and standard deriva-tion of recognition accuracies are estimated over 5 random splits of the dataset.

We first conduct one test to demonstrate the positiveness exclusive property of PER term $\Omega(\tilde{\alpha})$. For both m-LapSVM and PER regularized m-LapSVM, we evaluate the classifiers learnt at each iteration step of optimization on the unlabeled data, and count and plot the number of unlabeled samples that receive positive scores from more than one classifiers. Figure 2 shows the obtained evolving curves. In this test, we initialize $\tilde{\alpha}_0$ with an all-one vector, and set the maximum iteration number as $1,000$. From the curves, we can observe that, minimizing the PER term tends to reduce the number of unlabeled samples that receive positive scores from more than one classifiers. This test clearly demonstrates the effect of of $\ell_{1,2}$-norm type regularizer $\Omega(\tilde{\alpha})$ for inducing the proposed positiveness exclusion prior. The improvements brought by PER in terms of classification accuracy will be evaluated in the rest part of this section.

Figure 3 shows the average accuracies and standard deviations of various methods while using different number of training images. From this figure we can see that m-LapSVMs regularized with PEP term consistently outperforms m-LapSVMs and m-SVMs under different training settings. In details, for the usage of 5 training images per subject, McSVM achieves the recognition accuracy of 67.50%, which is boosted to 70.69% by using Laplacian regularizer, and further increased with 4.63 percents if additionally imposing the PEP term. These results with comparisons demonstrate that proposed PEP term can well guide the transductive learning procedure to improve recognition accuracy.

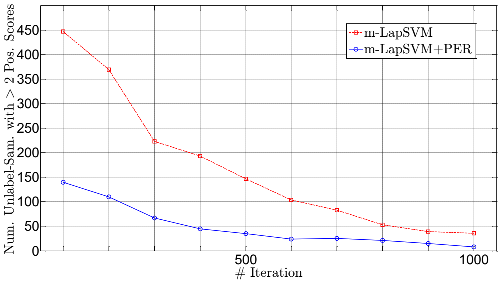

Figure 2. Positiveness exclusion inducing property of PER. X-axis: iteration steps of optimization; Y-axis: number of unlabeled samples that receive positive scores from more than two classifiers. See texts for more details.

4.3. Exp-II: Results on Scene-15 Dataset

Scene-15 dataset [24] is composed of 15 scene classes. Each class contains 200 to 400 images and there are 4485 images in total. As suggested [15, 22], we randomly select 50 or 100 images from each category for model training and use the rest images for test. We also follow the image classification architecture proposed in [21] as introduced in last experiment. Note that while extracting image descriptors, we construct a 1024-dimension visual dictionary as in previous works [15, 22]. We perform PCA on the extracted 1024-dimension features as a pre-processing step to obtain the lower-dimension image features (to keep 98% energy). A 4 cross-validation procedure is conducted on the training subset to determine the free parameters in Algorithm 1.

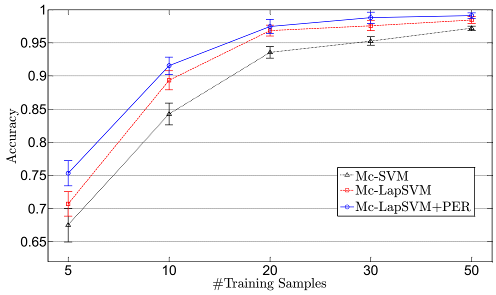

Figure 3. Recognition accuracies of various methods on YALEB dataset [9].

We report the related evaluation results in Table **??**. The results of SPM [15] and ScSPM [22] are also given for comparison. While using 100 training samples pe class, the improvements brought by PER regularized m-LapSVMs method over m-LapSVMs and m-SVMs are 1.27 and 3.30 percents respectively as shown Table **??**. For the usage of 50 training samples per class, we obtain the margins of 1.66 and 4.39 percents over m-LapSVMs and m-SVMs respectively. These results demonstrate that our proposed SVM formulation with PER can improve the classification performance on this dataset.

Table 1. Classification accuracy (%) comparison on Oxford Flower-102 dataset.

Algorithms	Accuracy
NS [23]	46.6
MTJSRC [23]	54.7
m-SVMs	55.70
m-LapSVMs	57.97
m-LapSVMs with PER	**59.41**

4.4. Exp-III: Results on Oxford Flower-102 Dataset

Flower-102 dataset [18] consists of 8189 images divided into 102 flower classes. Each class contains 40-250 images. As suggested in [18], we collect 10 images per class for training, other 10 images per class for validation, and use the rest images for testing. Following the standard protocol [18], we use the SIFTint feature provided by the dataset and compute kernel matrices as $\exp(-\mathcal{X}^2(x,x')/\mu)$ where μ is set to be the mean value of the pairwise $\mathcal{X}^2$ distance on the training set.

Table 1 summarizes the evaluation results of various algorithms. In addition to the three SVM-based methods, we also compare with two baseline methods, including the multi-task joint sparse representation based classifier (MTJSRC) [23] and the method based on nearest subspace algorithm [23]. For both methods, we directly use the results reported in [23]. From the table, we can see that our proposed PER term can boost the classification accuracies of m-LapSVMs and m-SVMs with the margins of 1.44 percents and 2.27 percents respectively. It is worthy to note that our proposed method can achieve higher accuracy than MTJSRC, which is also based on the MTL framework, in particular, with lasso-type regularizer. In contrast, our method adopts a max-structure $\ell_{1,2}$-norm regularizer, which can naturally induce the proposed positiveness exclusion prior.

5. Conclusions and Future Work

This paper presents a novel MTL framework along with its optimization for learning multi-class semi-supervised SVMs. We form the individual binary one-vs-rest S3VMs as learning tasks. These tasks are coupled by imposing the PER which induces positiveness exclusive prior on output scores of classifiers. The proposed method can be efficiently optimized with smoothing approximation technique. Extensive experiments on challenging visual classification benchmarks consistently validate the superior of our method to traditional way of learning m-S3VMs independently. In summary, we conclude with observations that it is beneficial to learn m-S3VMs jointly within a PER regularized MTL framework.

Our method can be extended by combining PER with: 1) convex empirical risks other than hinge loss, e.g., logistic loss, 2) semi-supervised SVMs other then LapSVMs, e.g., transductive SVMs [12], and 3) unsupervised learning or supervised learning with noisy / weak supervision. These extensions will be exploited in future work.

6. Acknowledgement

This work is done for CSIDM Project No. CSIDM-200803 partially funded by a grant from the National Research Foundation (NRF) administered by the Media Development Authority (MDA) of Singapore. This work is also supported by the China 863 program (Grant No.2008AA01Z126).

Appendix

1. Proof of Proposition 1

Proof. By definition we have that

$$\|[p]_+\|_1 = \sum_{i=1}^{n} \max\{0, p_i\} = \sum_{i=1}^{n} \max_{0\le v_i\le 1} p_i v_i = \max_{0\le v\le 1} \langle p, v\rangle, \tag{1}$$

where the second equality follows the fact that $\max\{0,a\} = \max_{0\le b\le 1} ab$. □

2. Proof of Proposition 2

Proof. Since $0 \in \{v_j : 0 \le v_j \le 1\}$, by definition we get that

$$0 \le q_{j,\mu}(\tilde{\alpha}) \le \max_{0\le v_j\le 1} \langle \tilde{K}_j\tilde{\alpha}, v_j\rangle = \|[\tilde{K}_j\tilde{\alpha}]_+\|_1. \tag{B.1}$$

Therefore

$$h_\mu(\tilde{\alpha}) = \frac{1}{2}\sum_{j=l+1}^{n} q_{j,\mu}^2(\tilde{\alpha}) \le \frac{1}{2}\sum_{j=l+1}^{n} \|[\tilde{K}_j\tilde{\alpha}]_+\|_1^2 = h(\tilde{\alpha}). \tag{B.2}$$

Since $0 < v_j \le 1$ we have $\|v_j\|^2 \le M$. Therefore,

$$q_{j,\mu}(\tilde{\alpha}) \ge \max_{0\le v_j\le 1} \langle \tilde{K}_j\tilde{\alpha}, v_j\rangle - \frac{\mu M}{2} = \|[\tilde{K}_j^T\tilde{\alpha}]_+\|_1 - \frac{\mu M}{2}. \tag{B.3}$$

By (B.1) and (B.3) we get

$$|q_{j,\mu}(\tilde{\alpha}) - \|[\tilde{K}_j\tilde{\alpha}]_+\|_1| \le \frac{\mu M}{2}. \tag{B.4}$$

Thus

$$\begin{aligned}
& |q_{j,\mu}^2(\tilde{\alpha}) - \|[\tilde{K}_j\tilde{\alpha}]_+\|_1^2| \\
= \;& |q_{j,\mu}(\tilde{\alpha}) - \|[\tilde{K}_j\tilde{\alpha}]_+\|_1| \cdot |q_{j,\mu}(\tilde{\alpha}) + \|[\tilde{K}_j\tilde{\alpha}]_+\|_1| \\
\le \;& \frac{\mu M}{2} 2\|[\tilde{K}_j\tilde{\alpha}]_+\|_1 \le \mu M \|\tilde{K}_j\tilde{\alpha}\|_1 \\
\le \;& \mu M^2 R \|\tilde{K}_j\|_1.
\end{aligned} \tag{B.5}$$

which implies that

$$q_{j,\mu}^2(\tilde{\alpha}) \ge \|[\tilde{K}_j\tilde{\alpha}]_+\|_1^2 - \mu M^2 R \|\tilde{K}_j\|_1. \tag{B.6}$$

By summarizing both sides of the above inequality over j from $l+1$ to n, we immediately get

$$h_\mu(\tilde{\alpha}) \ge h(\tilde{\alpha}) - \mu M^2 R \sum_{j=l+1}^{n} \|\tilde{K}_j\|_1. \tag{B.7}$$

Combining (B.1) and (B.7) we get (8). □

3. Proof of Theorem 1

Proof. From the standard results (see, e.g.,[16, Theorem 1]) we have that $q_{j,\mu}(\tilde{\alpha})$ is well defined and continuously differentiable, and its gradient

$$\nabla q_{j,\mu}(\tilde{\alpha}) = \tilde{K}_j^T v_j(\tilde{\alpha}) \tag{C.1}$$

is Lipschitz continuous with constant

$$L_{j,\mu} = \frac{\|\tilde{K}_j\|_2^2}{\mu}. \tag{C.2}$$

Since $h_\mu(\tilde{\alpha})$ is the summation of the *squares* of smooth functions $q_{j,\mu}(\tilde{\alpha})$, it is also well defined with gradient

$$\nabla h_\mu(\tilde{\alpha}) = \sum_{j=l+1}^{n} q_{j,\mu}(\tilde{\alpha})(\tilde{K}_j^T v_j(\tilde{\alpha})). \tag{C.3}$$

To prove the Lipschitz continency of $\nabla h_\mu(\tilde{\alpha})$, we first show the Lipschitz continuousness of $q_{j,\mu}(\tilde{\alpha})\nabla q_{j,\mu}(\tilde{\alpha})$, or equivalently that of $q_{j,\mu}(\tilde{\alpha})\nabla q_{j,\mu}(\tilde{\alpha})$:

$$\begin{aligned}
& \|q_{j,\mu}(\tilde{\alpha}_1)\nabla q_{j,\mu}(\tilde{\alpha}_1) - q_{j,\mu}(\tilde{\alpha}_2)\nabla q_{j,\mu}(\tilde{\alpha}_2)\|_2 \\
= \; & \|q_{j,\mu}(\tilde{\alpha}_1)\nabla q_{j,\mu}(\tilde{\alpha}_1) - q_{j,\mu}(\tilde{\alpha}_1)\nabla q_{g,\mu}(\tilde{\alpha}_2) \\
& +q_{j,\mu}(\tilde{\alpha}_1)\nabla q_{j,\mu}(\tilde{\alpha}_2) - q_{j,\mu}(\tilde{\alpha}_2)\nabla q_{j,\mu}(\tilde{\alpha}_2)\|_2 \\
\leq \; & |q_{j,\mu}(\tilde{\alpha}_1)| \cdot \|\nabla q_{j,\mu}(\tilde{\alpha}_1) - \nabla q_{j,\mu}(\tilde{\alpha}_2)\|_2 \\
& +\|\nabla q_{j,\mu}(\tilde{\alpha}_2)\|_2 \cdot |q_{j,\mu}(\tilde{\alpha}_1) - q_{j,\mu}(\tilde{\alpha}_2)| \\
\leq \; & \left(\frac{\|\tilde{K}^j\|_2^2\|\tilde{K}_j\|_1 MR}{\mu} + \|\tilde{K}_j\|_2^2 M\right)\|\tilde{\alpha}_1 - \tilde{\alpha}_2\|_2 \\
= \; & M\|\tilde{K}_j\|_2^2\left(\frac{\|\tilde{K}_j\|_1 R}{\mu} + 1\right)\|\tilde{\alpha}_1 - \tilde{\alpha}_2\|_2 \qquad \text{(C.4)}
\end{aligned}$$

where the last inequality follows the basic facts: (i) (C.2), (ii) $q_{j,\mu}(\tilde{\alpha}_1) \leq \|[\tilde{K}_j\tilde{\alpha}_1]_+\|_1 \leq \|\tilde{K}_j\tilde{\alpha}_1\|_1 \leq \|\tilde{K}_j\|_1 MR$, (iii) $\|\nabla q_{j,\mu}(\tilde{\alpha}_2)\|_2 = \|\tilde{K}_j v_j(\tilde{\alpha}_2)\|_2 \leq \|\tilde{K}_j\|_2\sqrt{M}$, and (iv) $|q_{j,\mu}(\tilde{\alpha}_1) - q_{j,\mu}(\tilde{\alpha}_2)| \leq \|\tilde{K}_j\|_2\sqrt{M}\|\tilde{\alpha}_1 - \tilde{\alpha}_2\|_2$ (due to the boundness of $\nabla q_{j,\mu}$ in (iii)). By combining (C.3) and (C.4) we get the validity of (9). □

References

[1] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73 (3):243–272, 2008.

[2] M. Belkin and P. Niyogi. Using manifold stucture for partially labeled classification. In *NIPS*.

[3] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *JMLR*, 2004.

[4] K. Bennett and A. Demiriz. Semi-supervised support vector machines. In *NIPS*, 1999.

[5] R. Caruana. Multi-task learning. *Machine Learning*, 28:41–75, 1997.

[6] O. Chappelle. Training a support vector machine in the primal. *Neural Computation*, 19(5):1155–1178, 2007.

[7] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *JMLR*, 2:265–292, 2001.

[8] T. Evgeniou and M. Pontil. Regularized multi–task learning. In *SIGKDD*, 2004.

[9] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE TPAMI*, 23(6):643–660, 2001.

[10] C.-W. Hsu and C.-J. Lin. A comparison of methods for multiclass support vector machines. *TNN*, 13:415–425, 2002.

[11] T. Joachims. Transductive learning via spectral graph partitioning. In *ICML*.

[12] T. Joachims. Transductive inference for text classification using support vector machines. In *ICML*, 1999.

[13] G. S. Kimeldorf and G. Wahba. A correspondence between bayesian estimation on stochastic processes and smoothing by splines. *Annals of Mathematical Statistics*, 41:495–502, 1970.

[14] M. Kowalski. Sparse regression using mixed norms. *Applied and Computational Harmonic Analysis*, 27(3):303–324, 2009.

[15] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[16] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

[17] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39:103–134, 2000.

[18] M. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *ICCVGIP*.

[19] G. Obozinski, B. Taskar, and M. Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Journal of Statistics and Computing*, 20:231–252, 2009.

[20] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. *submitted to SIAM Journal of Optimization*, 2008.

[21] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classificataion. In *CVPR*, 2010.

[22] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

[23] X. Yuan and S. Yan. Visual classification with multi-task joint sparse representation. In *CVPR*, 2010.

[24] H. Zhang, A. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *CVPR*, 2006.

[25] Y. Zhou, R. Jin, and S.-C. Hoi. Exclusive lasso for multi-task feature selection. In *AISTAS*, 2010.

[26] X. Zhu. Semi-supervised learning literature survey. 2008.